AppleScript 輔助說明

要點

關於 AppleScript

在使用 Mac 時，您會發現自己時常重複執行一些工作。使用 AppleScript 可讓您編寫工序指令和應用程式，來將這些重複工作自動化。

AppleScript 是功能強大且靈活彈性的工序指令語言，內建於 Mac OS X 中。您可以利用 AppleScript 來建立快速鍵、自動化重複性工作，甚至自定應用程式以節省大量時間。AppleScript 能讓您控制可工序化的應用程式。

AppleScript 工序指令包含語法接近英語的指令，可以在物件上執行一些動作。有些動作範例為開啟、關閉和取得名稱等。物件的部分範例為應用程式、視窗或是偏好設定。

工序指令可以控制一個或多個應用程式。若要成為可工序化的程式，該應用程式必須要定義好可供工序指令使用的術語。Mac OS X 隨附了數個可工序化的應用程式。

工序指令也可以透過網路控制應用程式，所以您可以將多台電腦上的工作自動化。

事實上，您已經開始使用 AppleScript 而不自知。系統與應用程式中有許多功能其實都是從應用程式之中所呼叫的 AppleScript 工序指令。例如，在此輔助說明系統中，每當您按下連結打開應用程式或檔案時，它都是個執行動作的 AppleScript 工序指令。

相關主題

AppleScript 的運作方式
關於 AppleScript 編寫程式
關於 AppleScript 語言
Internet 上的 AppleScript 資源
AppleScript Studio 開發環境

要點

關於 AppleScript 編寫程式

“AppleScript 編寫程式”是個易於使用的工序指令應用程式，可供讀取、撰寫、錄製和儲存 AppleScript 工序指令。

這些好用的“AppleScript 編寫程式”功能如下：

- 提供一致且更容易使用的介面
- 讓您在工序指令中尋找和取代文字
- 自動對過長的工序指令敘述句進行折行
- 允許您執行多次的還原動作
- 提供快速鍵選單工序指令，使工序指令編寫更快速
- 支援應用程式服務

- 可讓您使用“程式庫面板”來取用應用程式的工序指令
- 提供“工序指令輔助程式”來幫助您完成敘述句
- 提供結果總覽和 Event 記錄總覽，來讓您檢查工序指令中進行的項目

相關主題

關於 AppleScript 語言
AppleScript Studio 開發環境

要點

製作 AppleScript 工序指令

您不需知道任何有關程式設計方面的知識，就可以享受 AppleScript 所帶來的好處。事實上，您甚至不用為工序指令撰寫任何一個字。Mac OS X 中內含上百個實用的 AppleScript 可供您立即開始使用，而且在網路上還可以找到上千個 AppleScript 工序指令。

若您想要更加深入，可以使用既有的工序指令並依照您的需求稍加修改。

當然，製作您自己的工序指令對您最為有利。學習如何撰寫工序指令可以讓您製作實用的工具和節省時間的自動化功能，讓您工作更有生產力。

您可以使用“AppleScript 編寫程式”來修改或製作您自己的工序指令，它提供簡單的環境供您編寫工序指令。“AppleScript 輔助說明”中所有範例都是使用“AppleScript 編寫程式”完成的。

- 打開“AppleScript 編寫程式”。

  在“AppleScript 編寫程式”中輸入以下文字。

  ```
  say "Welcome to AppleScript!"
  ```

  按一下“執行”。

相關主題

關於 AppleScript 編寫程式
AppleScript Studio 開發環境

要點

AppleScript 的運作方式

AppleScript 藉由傳送訊息來運作，稱為“Apple Event”。當您執行工序指令，AppleScript 將 Apple Event 傳送給執行指令的系統或應用程式。

在大部分的情況下，應用程式或系統會將結果回傳給 AppleScript。例如，若工序指令送出一個要求給應用程式以取得物件的名稱，則應用程式就會回傳該名稱。如果指令無法執行，則應用程式或 AppleScript 會傳回錯誤訊息。您可以在“AppleScript 編寫程式”中看到傳回的結果。

一般而言，您不需要瞭解 Apple Event 的機制。AppleScript 會在背景進行這一切。您只需要撰寫並執行您的工序指令。

相關主題

關於 AppleScript
關於 AppleScript 語言

要點

關於 applet

儲存為應用程式的工序指令稱為 AppleScript 應用程式或 applet。Applet 會在開啟時自動執行，並執行工序指令中的指令。Applet 可以是僅包含幾個指令的小型函數，或是包含上千行指令的複雜應用程式。

Applets 可以儲存成可編輯格式或儲存為“只供執行”。可編輯的 applet 仍然保留工序指令的文字，所以它可以隨時編輯並重新編譯。若要編輯或更改 applet，請將它拖移至“AppleScript 編寫程式”圖像上面。當您將工序指令儲存為“只供執行”時，則移除可編輯的文字部分，而檔案再也不能被編輯或重新編譯。

若要編輯已儲存為可編輯格式的 applet，您可以在“AppleScript 編寫程式”或其他 AppleScript 編寫工具中打開它。Applet 需要有個“on run”的處理程式，可以在打開 applet 時開始執行工序指令。

- 此範例顯示一個“on run”處理程式。

```
on run
        --first command
        --second command
end run
```

相關主題

關於 droplet
AppleScript 檔案類型

要點

關於已編譯的工序指令

已編譯的工序指令是個以可編輯形式儲存的 AppleScript 檔案，您可以在“AppleScript 編寫程式”或其他 AppleScript 編寫程式中讀取或編輯它。

當您打開已編譯的工序指令時，它會顯示在“AppleScript 編寫程式”中，但不會自動執行任何指令。這可讓您不用實際執行工序指令即可修改它。再來您可以編譯工序指令，即在檔案中讀取所有文字並將它們轉換成 AppleScript 程式碼。已編譯的工序指令可被執行。已編譯的工序指令的檔案副檔名為 .scpt。

當您將檔案儲存為已編譯的工序指令時，其文字仍然與程式碼一起儲存在檔案中。因此，已編譯的工序指令可隨時再度編輯。

相關主題

關於 applet
關於 droplet
AppleScript 檔案類型

要點

關於 droplet

droplet 是當您將檔案或檔案夾拖放到它的圖像上時執行的 AppleScript 工序指令。此工序指令在檔案或檔案夾上執行動作。

droplet 就跟 applet 一樣，能儲存為可編輯格式或是唯讀格式。可編輯的 droplet 仍然保留工序指令的文字，所以它可以隨時編輯並重新編譯。若要編輯或更改 droplet，請將它拖移至“工序指令編寫程式”圖像上面。當您將工序指令儲存為“只供執行”時，則移除可編輯的文字部分，而檔案再也不能被編輯或重新編譯。

Droplets 需要有個“on open”的處理程式，可以在打開 droplet 時開始執行工序指令。“on open”處理程式有個參數包含拖移至 droplet 的項目列表。droplet 工序指令的一部分會檢查列表，另一部分對該項目產生作用。除非 droplet 設計只在單一項目上執行動作，否則它應包含一個重複迴圈反覆使用每一個項目，依序在每一個項目之上執行動作。

- 以下為帶有“on open”droplet 處理程式和參數列表的範例。

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

相關主題

關於 applet
關於已編譯的工序指令
AppleScript 檔案類型

要點

使用 AppleScript 指令詞彙

若想知道可以用來控制應用程式的指令與物件，可以查閱該應用程式的指令詞彙。指令詞彙包含一組應用程式瞭解的相關術語。

您可以在“程式庫”視窗中選擇應用程式，或是將應用程式的圖像拖至“AppleScript 編寫程式”的圖像上，以便在“AppleScript 編寫程式”檢視應用程式的指令詞彙。

指令詞彙中的術語是以“套件”作為名稱整理成相關的術語群組。指令和物件會以粗體文字顯示。

1. 在“AppleScript 編寫程式”中，選擇“視窗”>“程式庫”。
2. 在“程式庫”視窗中的應用程式名稱上按兩下。
3. 選擇一個套件。

   顯示此套件的可用字彙與其說明。

AppleScript 語言分成好幾種不同的類別：

- 套件：相關術語的群組。在“AppleScript 編寫程式指令詞彙”中，套件顯示在第一欄中，以橘色大寫字母“S”表示。
- 指令：要求動作或結果。指令，又稱為“事件”，顯示在第二欄中，以藍色大寫字母“C”（也稱為一個事件）表示。
- 類別：支援工序指令的物件。類別說明列出類別中已存在的屬性與元件。類別顯示在第二欄，以淺紫色大寫字母“C”表示。

- 屬性：內含值的物件。屬性顯示在第三欄，以紫色大寫字母“P”表示。
- 元件：包含在其他物件之中的物件。例如，段落可能包含許多字。元件顯示在第三欄，以橘色大寫字母“E”表示。

相關主題

類別定義
指令定義
使用 AppleScript 來以指令操作應用程式

打開

AppleScript Language Guide（AppleScript 語言指南）：指令參考
AppleScript Language Guide（AppleScript 語言指南）：類別參考

要點

解決問題

若您在使用 AppleScript 時發生問題，請參閱下列主題中的資訊來解決問題。

當使用者按下“從列表中選擇”對話框的“取消”時，出現錯誤訊息

如果您的檔案夾動作非預期地自動啟用

若您的工序指令無法辨識磁碟

基本指令

類別定義

類別定義敘述了可支援工序指令之物件的屬性和元件。類別可比喻為句子內的名詞；類別定義就像字典內名詞的定義。

一般來說，AppleScript 指令詞彙中的類別定義是以術語起首，以粗體字表示，後面接的是在同一行內的簡短術語描述。

後面接著的是屬於或包含支援工序指令之物件的任何元件或物件列表。未必所有的類別定義裡都包含元件列表，因為有些可工序化物件有包含其他可工序化物件，但有些是不包含的。

類別定義會以可工序化物件或數值的屬性或品質列表作為結尾。

屬性項目首先是屬性的名稱，後面接著：

- 資料格式的描述，其中儲存了屬性的數值，例如：整數（數字）、小數點（兩組數字的列表）、字串（文字資料）和 Boolean（true 或 false）等等
- 選擇性類型指示器（[r/o]），指出屬性值是“唯讀”（只能讀取不能更改）
- 屬性的簡短描述

如果物件有複數形，則表示您可以參照多個物件。

如果物件具有元件列表，則表示該物件含有其他物件。元件名稱（可用於工序指令中）以粗體顯示。以普通體顯示的文字是用以說明參考元件的方法。

物件可能會包含屬性。屬性名稱是以粗體顯示。以普通體顯示的文字則說明該屬性值包含的類型（如：字串）。屬性可以是常數或常數列表中的一個數值。除非該屬性語法的後面不含“[r/o]”（唯讀），工序指令可以取得並設定屬性的數值。

相關主題

指令定義
物件定義
AppleScript 指令
使用 AppleScript 指令詞彙

打開

AppleScript Language Guide（AppleScript 語言指南）：類別參考

指令定義

指令就是 AppleScript 中負責執行動作的詞彙。AppleScript 指令詞彙中的指令都顯示在中央欄位，並以藍色大寫字母“C”表示。指令的定義包含了對指令功能和其語法的簡述。語法是您在工序指令裡編寫指令時應該使用的方式。語法可分成幾個部分。

粗體字是在敘述句裡使用的術語，必須與指令詞彙裡所使用的完全相同。

藍色文字部分是顯示您提供的資訊。例如，您可以辨識已定義的物件或變數。

指令中可能有些選擇性的部分，這些部分會出現在括號內。您並不一定要在敘述句中使用這些指令。

相關主題

類別定義
使用 AppleScript 指令詞彙
事件定義
物件定義

打開

AppleScript Language Guide（AppleScript 語言指南）：指令參考

事件定義

事件定義敘述了在支援工序指令的物件中，可用來選定目標動作的動詞或指令。event 是物件能完成或參與的動作或動詞。

一般來說，event 定義是以術語開始，後面接著對該術語的簡短描述（位於同一行內）。

後面繼續接的是用於執行指令的參數列表。選擇性的參數會包含在括號內；必要參數則沒有。

參數項目會以參數的名稱作為開始，名稱後面接著參數使用的資料格式描述，最後是以簡短的參數描述作為結尾。

指令的定義包含了對指令功能和其語法的簡述（加底線的文字）。語法是您在工序指令裡編寫指令時應該使用的方式。語法可分成幾個部分。

粗體字是在敘述句裡使用的術語，必須與指令詞彙裡所使用的完全相同。

普通體文字部分是顯示您提供的資訊。例如，您可以辨識已定義的物件或變數。

指令中可能有些選擇性的部分，這些部分會出現在括號內。您並不一定要在敘述句中使用這些指令。

指令會對物件產生作用。物件的定義會顯示辨識物件的語法，包括名稱以及簡述（加底線的文字）。物件的類別會定義某一物件的特性，例如可在該物件上使用的指令。

相關主題

資料庫事件
類別定義
指令定義
物件定義
AppleScript 的運作方式

基本指令

物件定義

指令會對物件產生作用。AppleScript 指令詞彙中的物件定義會顯示辨識物件的語法，包括名稱以及簡述（加底線的文字）。物件的類別會定義某一物件的特性，例如可在該物件上使用的指令。

如果物件有複數形，則表示您可以參照多個物件。

如果物件具有元件列表，則表示該物件含有其他物件。元件名稱（可用於工序指令中）以粗體顯示。以普通體顯示的文字是用以說明參考元件的方法。

物件可能會包含屬性。屬性名稱是以藍色粗體字顯示。以藍色普通體顯示的文字則說明該屬性值包含的類型（如：字串）。屬性可以是常數或常數列表中的一個數值。除非該屬性語法的後面不含“[r/o]”（唯讀），工序指令可以取得並設定屬性的數值。

相關主題

使用 AppleScript 來以指令操作應用程式
指令定義
類別定義
告訴物件要執行的動作
使用 AppleScript 指令詞彙

打開

AppleScript Language Guide（AppleScript 語言指南）：類別參考
AppleScript Language Guide（AppleScript 語言指南）：屬性

基本指令

If 敘述句

if 敘述句是一種指令，這個指令只有在符合特定條件時才會執行。每一個 if 敘述句包含一個 Boolean 表示式，其計算結果為 true 或 false。if 敘述句內含的指令只有在 Boolean 表示式的值為 true 時才會執行。if 敘述句的結尾是 end if。例如：

```
if amountReceived > 100 then
	set memberStatus to "special"
end if
```

在上列敘述句中，若 Boolean 表示式為 true（接收到的數量大於 100），則 if 敘述句中的指令就會執行。若 Boolean 表示式為 false，則指令不會執行。

if 敘述句還可能伴隨 else 敘述句。這樣的結合被稱為 if-then-else 敘述句。例如：

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

在上列敘述句中，若第一個 Boolean 表示式為 true（接收的數量大於 100），則第一部分敘述句中的指令就會執行。若第一個 Boolean 表示式為 false，則 else 之後第二部分敘述句中的指令會執行。

您可以在每個迴圈中置入巢狀的 if 敘述句來處理多個條件的情況。

若要瞭解更多關於 AppleScript 中 if 敘述句的相關資訊，請參閱 AppleScript Language Guide（AppleScript 語言指南）。

相關主題

Repeat 敘述句
指令定義
設定 AppleScript 中的變數
常數值

打開

AppleScript Language Guide（AppleScript 語言指南）- If

基本指令

Repeat 敘述句

您可以視情況使用 repeat 敘述句來多次執行作業。repeat 敘述句會持續在迴圈中執行一組敘述句，直到符合條件之後才離開迴圈。您可以在每個迴圈中置入巢狀的 repeat 敘述句來回應不同的情況。

AppleScript 有多種 repeat 敘述句，可回應不同的情況。

---

repeat (forever)

在迴圈中持續重複執行。使用 exit 敘述句離開迴圈。

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

---

repeat (number) times

依照所指定的次數重複執行。

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

repeat while

當指定條件為 true 時，則重複執行。

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

repeat until

重複執行至指定條件為 true 時。

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

repeat with (variable) from (startValue) to (stopValue)

設定變數值為 startValue，然後在每一次執行迴圈時遞增變數值。當變數遞增至 stopValue 時便停止重複迴圈。

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

repeat with (variable) in (list)

在列表的第一個項目中變數設定。每次通過迴圈後，會對下一個項目設定變數。當列表中所有項目皆已使用過時則終止迴圈的重複。

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

若要瞭解更多關於 AppleScript 中 repeat 敘述句的相關資訊，請參閱 AppleScript Language Guide（AppleScript 語言指南）。

相關主題

If 敘述句
設定 AppleScript 中的變數
常數值

打開

AppleScript Language Guide（AppleScript 語言指南）- repeat

基本指令

常數值

常數就是 AppleScript 中的保留字，包含事先決定的值。AppleScript 內含好幾個常數，您可以用來取得物件的相關資訊，執行比較，或進行數值計算。應用程式與其他物件可以定義自己的常數。這些常數都定義在 value class 的定義中。

若要檢視常數值，請將下列範例貼入“AppleScript 編寫程式”，然後按一下“執行”。按一下“結果”來檢視常數值。

結果也是常數。這是個工序指令常數，會參照指令所傳回的值。

```
get the current date
```

以下為部分其他常數範例：

- 數值常數：pi
- Boolean 常數：true, false
- 日期與時間常數：minutes, hours, days, weeks
- 字串常數：return, space, tab
- 工序指令常數：anything、current application、it、me、missing value、my、result

若要瞭解更多關於 AppleScript 中常數的相關資訊，請參閱 AppleScript Language Guide（AppleScript 語言指南）。

相關主題

設定 AppleScript 中的變數
物件定義
類別定義
事件定義

打開

AppleScript Language Guide（AppleScript 語言指南）- constants

基本指令

設定 AppleScript 中的變數

變數就是工序指令中建立的容器，用來儲存值。您可以在建立變數的時候指定它的名稱，這也稱為識別碼。當工序指令執行時，您可將值置於變數中、存取值，或者修改值。變數值僅在工序指令執行期間保留。當工序指令結束，變數值會遺失。

- 這些範例顯示如何指定獨有的識別碼，並將數值指定到識別碼上。

```
set myVar to "153"

set x to "Quarterly Reports"
```

```
set x to false
```

若要檢視數值是如何儲存在變數中，請將這些範例貼入“AppleScript 編寫程式”，然後按一下“執行”。然後按“結果”一下，檢視變數中儲存的值。

若要瞭解更多關於 AppleScript 中變數的相關資訊，請參閱 AppleScript Language Guide（AppleScript 語言指南）。

相關主題

常數值
If 敘述句
Repeat 敘述句

打開

AppleScript Language Guide（AppleScript 語言指南）- 變數

基本指令

指定應用程式的位置

Mac OS X 中的應用程式是包裝成“套件”的型式。套件是一種特殊的目錄，其中包含組成應用程式的檔案夾和檔案。正因如此，“path to”和“info for”指令可能會對應用程式傳回不同的結果。

如果您使用應用程式的“path to”指令，路徑通常會以“.app”結尾。副檔名可能不會在 Finder 中顯示為檔名的一部分，但是您仍可使用副檔名來識別工序指令中的檔案。

如果您使用“info for”指令來取得關於應用程式的資訊，則結果的“檔案夾”屬性通常為“true”。您可以使用結果的其他屬性來取得關於檔案及其名稱的更多資訊；例如，隱藏的副檔名、副檔名或顯示的名稱。

相關主題

path to
info for
參考檔案夾、檔案和磁碟
choose application
Finder 工序指令

Finder 指令

參考檔案夾、檔案和磁碟

AppleScript 可讓您利用沒有包含路徑的名稱，參考某些檔案夾、檔案和磁碟。AppleScript 可識別以下項目的位置，如“文件”檔案夾、“桌面”、“個人專屬檔案夾”和“啟動磁碟”。

- 此範例顯示如何在工序指令中使用其名稱來參考特殊的檔案夾 。

```
set x to the path to the desktop folder
```

相關主題

在工序指令中參考 Finder
path to

info for
Finder 工序指令

Finder 指令

在工序指令中參考 Finder

若要使用 Finder 指令，您必須利用 tell 敘述句對 Finder 下命令，就像您對應用程式下命令一般。

- 以下的範例在工序指令中使用“tell”敘述句。tell 敘述句會將視窗功能的數量導向“Finder.app”。

```
tell application "Finder.app"
        return the number of windows
end tell
```

相關主題

Finder
告訴物件執行動作
參考檔案夾、檔案和磁碟
物件定義
Finder 工序指令

打開

AppleScript Language Guide（AppleScript 語言指南）- 物件指定元

Finder 指令

Finder 工序指令

您可以在 Finder 中使用下列指令：

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

參考檔案夾、檔案和磁碟

在工序指令中參考 Finder

reveal

select

set

sort

trash

update

window

標準工序指令附加程式指令

工序指令附加程式加入了許多有用的指令，讓您可以包含在工序指令中使用。

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write